KERMIT: logicalization of BERT

revised version 2.0

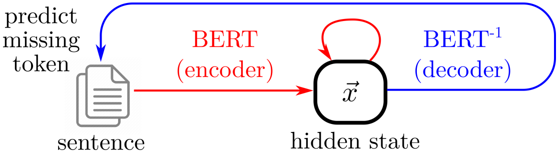
YKY

August 17, 2020

Table of contents

BERT's ground-breaking significance: closed-loop training

- BERT uses ordinary text corpuses to induce knowledge, forming representations that have universality:



(1)

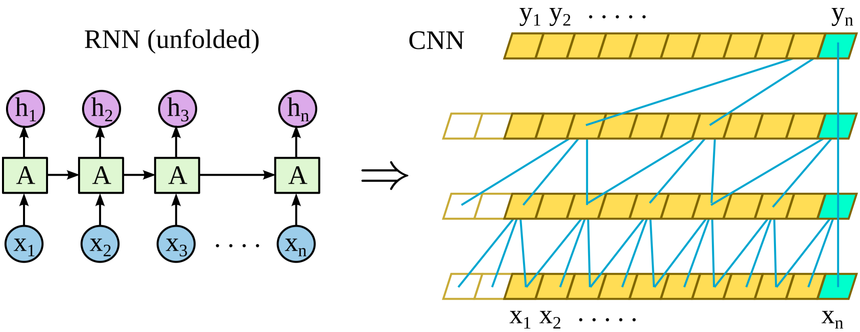
In other words, the hidden state compresses the meaning of sentences, that can be used in other scenarios

- This implies that human-level AI can be *induced* from existing corpora, without the need to retrace human infant development
- This training technique came from an earlier paper, unrelated to BERT's internal architecture

BERT's internal architecture

BERT results from combining several ideas:

- BERT is basically a seq-to-seq transformation
- Seq-to-seq was originally solved by RNNs
- But RNNs are slow, researchers proposed to replace them with CNNs



(2)

- CNN with attention mechanism gives rise to Transformer
- My idea is to incorporate logical symmetry into BERT while following this line of thinking

Symmetry in logic

- Words form sentences, analogous to concepts forming propositions in logic
- From an abstract point of view, logic can be seen as an algebra with 2 operations: a non-commutative multiplication ($\cdot$, for composition of concepts) and a commutative addition ($\wedge$, for conjunction of propositions)
- For example:

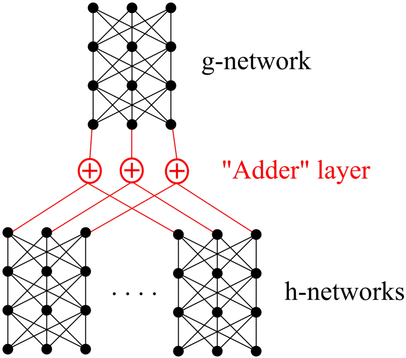
$$\begin{aligned} A \wedge B &\equiv B \wedge A \\ \text{it's raining} \wedge \text{lovesick} &\equiv \text{lovesick} \wedge \text{it's raining} \end{aligned} \tag{3}$$

- Word2Vec was also ground-breaking, but it was easy to go from Word2Vec to Sentence2Vec: just concatenate the vectors
  Sentences correspond to propositional logic
- A set of propositions requires symmetric NN to process, as elements of the set are permutation invariant

Symmetric neural network

- The symmetric NN problem has been solved by 2 papers:
  [PointNet 2017] and [DeepSets 2017]
- Any symmetric function can be represented by the following form (a special case of the Kolmogorov-Arnold representation of functions):

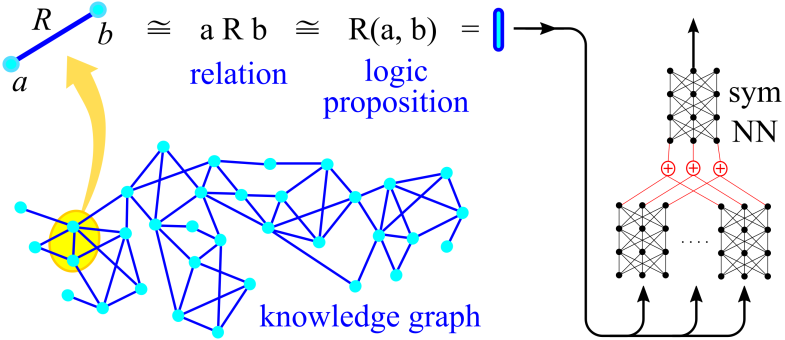
$$f(x, y, ...) = g(h(x) + h(y) + ...) \tag{4}$$



(5)

Knowledge graphs

- One cannot feed a knowledge graph directly into a neural network, as the input must be a vector. A solution is to break the graph into edges, where each edge is equivalent to a relation or proposition. One could say that graphs are isomorphic to logic

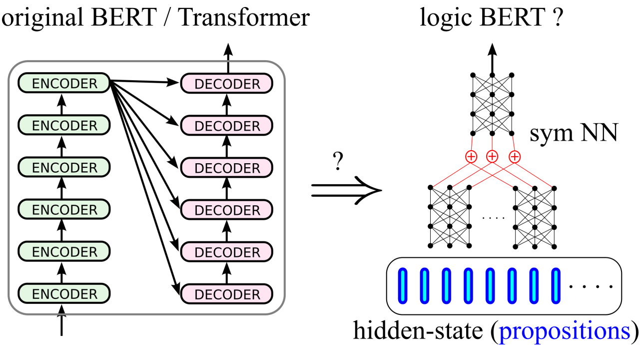


(6)

- As edges are invariant under permutations, it seems that we must use symmetric NNs to process them
- Logicalization provides a bridge between BERT and knowledge graphs Next we discuss BERT....

Logicalization of BERT

- We can force BERT's hidden state to be a set of propositions by imposing permutation symmetry on its Encoder:



(7)

- Below we'll see that BERT's attention mechanism is already symmetric and can perform logic inference

Connection between AI and logic

- If AI is based on logic, there must exist a precise connection between them
- BERT seems to be performing some kind of transformations between sentences, such sentences are simply compositions of word-embedding vectors:

$$\text{Socrates} \cdot \text{is} \cdot \text{human} \xrightarrow{BERT} \text{Socrates} \cdot \text{is} \cdot \text{mortal} \tag{8}$$

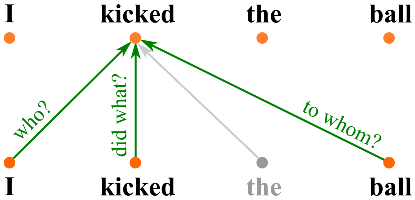
  While this may seem crude, it is effectively the same as a logic formula:

$$\forall x.\ \text{Human}(x) \to \text{Mortal}(x) \tag{9}$$

  Surprisingly, by the Curry-Howard correspondence, this formula corresponds to the mapping (8) above!
- In another set of slides we shall explore this connection. One could say the mathematical structure of logic is "eternal"; It will provide guidance for the long-term development of AI

What is attention?

- Attention originated with Seq2seq, then BERT introduced self-attention
- The essence of attention is weighing
- For each input, attention weighs the relevance of every other input and draws information from them accordingly to produce the output
- In BERT, attention is a relation among words in a sentence:

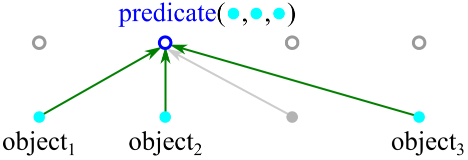


(10)

- From a logic point of view, words $\neq$ propositions
- In logic, the distinction between sub-propositional and propositional levels is of crucial importance!

Predicates vs propositions

- The word "predicate" comes from Latin "to declare"
- In logic, a predicate is a declaration without a subject or object; In other words, it is a proposition with "holes"
- Proposition = predicate + objects
- Eg: Human(John), Loves(John, Mary)
- From the logic point of view, the output of attention is the fusion of a predicate with its objects:



(11)

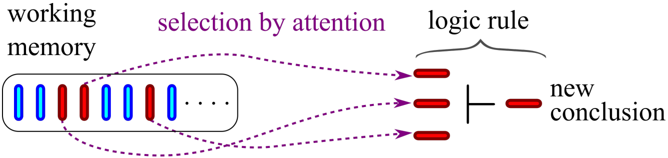
- Or figuratively:

$$\text{predicate} + \text{objects} \quad = \text{proposition}$$

(12)

“Attention is all you need” ?

- Analogously, attention on higher levels process relations among propositions
- We wish for attention to select propositions that are relevant for deduction:



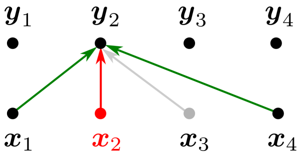
(13)

- But to choose K propositions from a set of N, there would be $\binom{N}{K}$ subsets, an exponential number
- BERT's way is to output only N propositions per each layer, each proposition is “supported” by all N propositions in the previous layer; The influence of premises are weighted by a matrix
- By the Curry-Howard isomorphism, BERT's mapping corresponds to some kind of alternative logic (BERT's creators may not have recognized this), which has very fast execution
- BERT's logic seems highly restricted, but the superficial restrictions may not prevent it from being a universal logic
- The key is to find a balance between speed and expressive power of the logic

- The simplest attention formula is: (where Q, K, V = query, key, and value matrices)

$$\boldsymbol{y}_j = \sum_i \langle Q\boldsymbol{x}_j, K\boldsymbol{x}_i \rangle V\boldsymbol{x}_i \tag{14}$$

(red indicates the focus of attention) This corresponds to a logic formula:



$$\equiv \quad \boldsymbol{x}_1 \wedge \boldsymbol{x}_2 \wedge \boldsymbol{x}_3 \wedge \boldsymbol{x}_4 \vdash \boldsymbol{y}_2 \tag{15}$$

In other words: $\boldsymbol{y}_j$ is the logic conclusion deduced from $\boldsymbol{x}_1, ..., \boldsymbol{x}_n$ with focus on $\boldsymbol{x}_j$

- "Focus" is not a logical concept; it is just a speed-up heuristic of BERT
- Easy to see that attention is permutation equivariant over $\boldsymbol{x}_1, ..., \boldsymbol{x}_n$, implying that its output are logic propositions, consistent with my theory
- We can also give a logical interpretation to multi-head attention: given the focus $\boldsymbol{x}_j$, there could be other premises leading to different conclusions:

$$\boldsymbol{x}_1 \wedge \boldsymbol{x}_2 \wedge \boldsymbol{x}_3 \vdash \boldsymbol{y}_1 \quad , \quad \boldsymbol{x}_1 \wedge \boldsymbol{x}_4 \wedge \boldsymbol{x}_5 \vdash \boldsymbol{y}_2 \tag{16}$$

Why is BERT so successful?

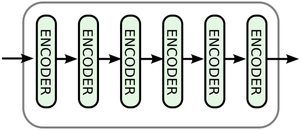
- The 6-layer BERT has $512 \times 6 = 3072$ heads, or 24576 with $8\times$ multi-head attention
- Each head does not simply correspond to 1 formula in conventional logic, may require further in-depth analysis....
- My guess is that the representation in BERT's higher layers are "high-level" propositions similar to the high-level features that represent complex objects in machine vision.
- The embedding of high-level propositions in vector space may be "semantically dense", meaning that slight changes in the vector position may convey many different meanings
- Because logic rules are organized in 6 layers of hierarchy, this structure has the "deep learning" property

A possible improvement of BERT

- The BERT attention formula (14) has some unnecessary restrictions, where generally we just need a symmetric function in the $\boldsymbol{x}_i$'s
- The general form of symmetric functions is given by (4)
- Immitating BERT, we introduce a "focus" of attention on $\boldsymbol{x}_j$:

$$\boldsymbol{y}_j = g(\ h(\boldsymbol{x}_j, \boldsymbol{x}_1) + ... + h(\boldsymbol{x}_j, \boldsymbol{x}_n)\ ) \tag{17}$$

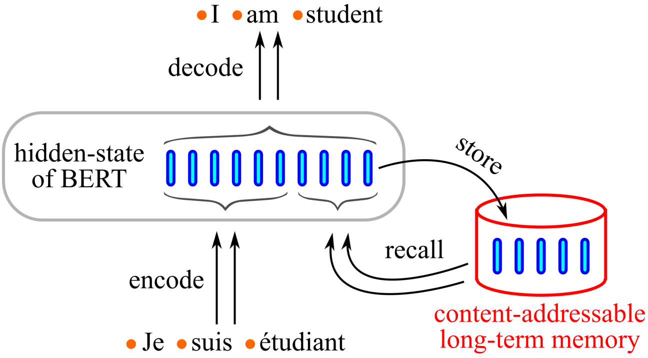
  this preserves equivariance
- We can use this function to replace the entire BERT Encoder:



(18)

Content-addressable long-term memory

- The content-addressable memory idea came from Alex Graves *et al*'s Neural Turing Machine [2014]
- The original BERT's hidden state lacked a logical structure; It was not clear what it contains exactly. With logicalization, propositions inside BERT can be stored into a long-term memory:



(19)

- Name: Knowledge-Enhanced Reasoning with Memorized Items
- This is getting very close to strong AI, and depends crucially on logicalization

- Eg. “The sun is hot”, “Water flows downhill” are facts that stay constant
- But some of this knowledge is implicitly stored as rules (matrix weights)
- Some knowledge is explicit, eg: “cats are mammals”, “smoking can cause cancer”
- Logicalization provides a way to interpret logic rules (weights)
- We can also store rules (weights) into content-addressable memory

Doubts about logicism

- Many people question: Do our brains really use symbolic logic to think?
- To say the least, all our languages are essentially in logical form
- Our impression is that the brain constructs "mental models" of the world and "reads off" conclusions from such models
- Consider a description: "Wife cheats on husband, stubs him with knife"

(20)

- What is she wearing? What color is her dress? Such details are imagined and unwarranted
- So what kind of details can our model have? The answer is: it cannot have ANY detail, except those entailed or constrainted by logic
- Models may be constructed from abstract logic propositions; Models with a lot of sensory details are implausible
- Perhaps the brain is much closer to formal logic than we'd thought

References

Questions, comments welcome ☺️

[1] Alex Graves, Greg Wayne, and Ivo Danihelka. "Neural Turing Machines". In: *CoRR* abs/1410.5401 (2014). arXiv: 1410.5401. URL: http://arxiv.org/abs/1410.5401.

[2] Qi et al. "Pointnet: Deep Learning on Point Sets for 3D Classification and Segmentation". In: *CVPR* (2017). https://arxiv.org/abs/1612.00593.

[3] Zaheer et al. "Deep sets". In: *Advances in Neural Information Processing Systems* 30 (2017), pp. 3391–3401.